Use Mobile Center to quickly get the insights that your app users won't tell you

# Overview

Mobile Center brings together multiple services, commonly used by mobile developers, into a single, integrated product. You can build, test, distribute, and monitor your mobile apps and easily add backend cloud services to scale your app to millions of users on demand.

# Objectives

* Create a React Native Cross-platform project.
* Create a Mobile Center app.
* Integrate the React Native project with the Mobile Center app.
* Track app analytics and events to Mobile Center.

# Prerequisites

* Mac OS X
* XCode
* A Mobile Center account (<https://mobile.azure.com>)

# Intended Audience

This Quick Start Challenge is intended for developers who are familiar with JS and React JS development.

# Task 1: Create a React Native app

You will need Node, Watchman, the React Native command line interface, and XСode.

We recommend installing Node and Watchman using [Homebrew](http://brew.sh/).

1. Run the following commands in a Terminal after installing Homebrew:

*brew install node*

*brew install watchman*

If you have already installed Node on your system, make sure it is version 4 or newer.

[Watchman](https://facebook.github.io/watchman) is a tool by Facebook for watching changes in the filesystem. It is highly recommended you install it for better performance.

Node comes with npm, which lets you install the React Native command line interface.

1. Run the following command in a Terminal:

*npm install -g react-native-cli*

1. The easiest way to install Xcode is via the Mac App Store. Installing Xcode will also install the iOS Simulator and all the necessary tools to build your iOS app. If you have already installed Xcode on your system, make sure it is version 8 or higher. You will also need to install the Xcode Command Line Tools. Open Xcode, then choose "Preferences..." from the Xcode menu. Go to the Locations panel and install the tools by selecting the most recent version in the Command Line Tools dropdown.
2. Testing your React Native Installation

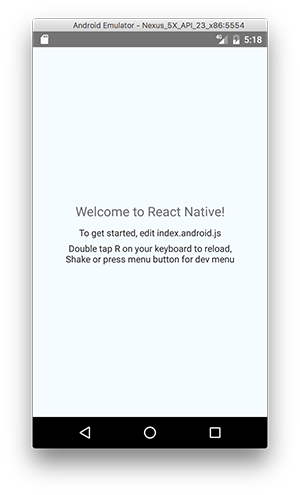
Use the React Native command line interface to generate a new React Native project called "AwesomeProject", then *run react-native run-ios* inside the newly created folder:

*react-native init AwesomeProject*

*cd AwesomeProject*

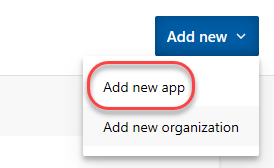
*react-native run-ios*

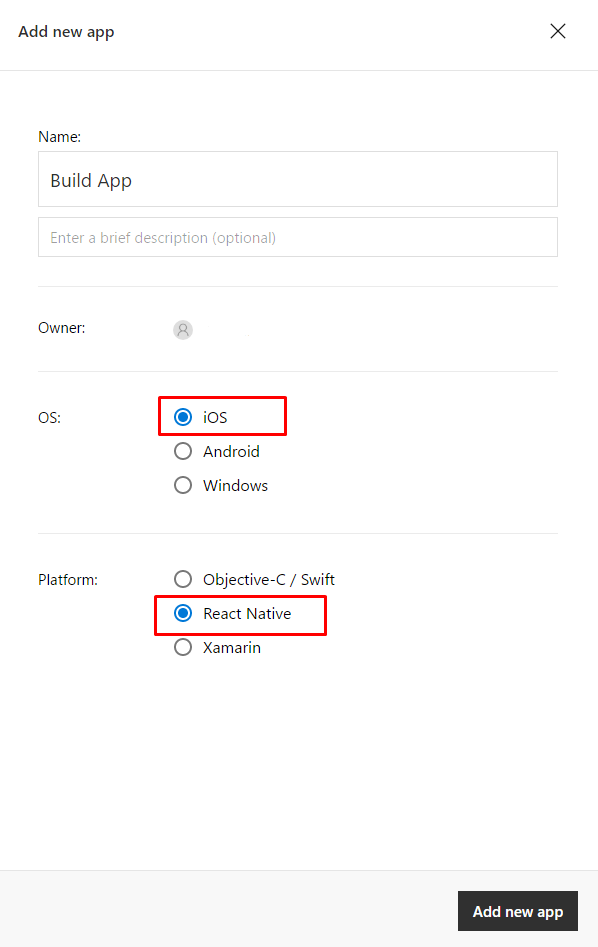
You should see your new app running in the iOS Simulator shortly.



# Task 2: Create the Mobile Center app

1. Log in to your Mobile Center account at <https://mobile.azure.com>.
2. From the top right corner, select **Add new | Add new app**.



1. Enter a **Name** of **“Build App”** and set it to be for **iOS** using **React Native**. Click **Add new app**. 
2. Once the new app has been created, there will be a set of instructions for integrating the app with Mobile Center. Note that the first step involves adding npm packages.



# Task 3: Integrate the React Native iOS app with Mobile Center

1. Return to projects’ folder and open **Terminal**, then run:

*npm install mobile-center-analytics –save*

*npm install mobile-center-crashes –save*

1. Link the plugin to the React Native app using the react-native link command:

*react-native link mobile-center-analytics*

*react-native link mobile-center-crashes*

**Note:** Linking the iOS SDK requires [CocoaPods](https://cocoapods.org/). If you do not have CocoaPods installed, follow these steps to link the SDK manually.

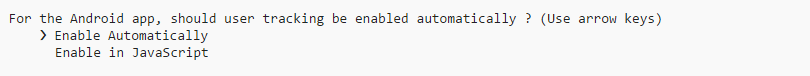
Download the [Mobile Center iOS SDK](https://github.com/Microsoft/MobileCenter-SDK-iOS/releases) and the [ReactNative Mobile Center iOS SDK](https://github.com/Microsoft/MobileCenter-SDK-React-Native/releases) frameworks provided as two zip files. Unzip the files and you will see different frameworks for each Mobile Center services and the ReactNative Mobile Center framework.

Add the binaries to your Xcode project:

* Make sure the Project Navigator is visible (⌘+1). Drag and drop the frameworks into your project to add them to the top level within your project.
* A dialog will appear. Select “Create groups” and set the checkmark for “Add to targets” for your target. Then click Finish.

**Note:** MobileCenter.framework is required to start the SDK. Make sure it is added to your project, otherwise the other modules won’t work and your app won’t compile.





# Task 4: Adding event tracking to the app

1. It may take a while for the analytics to begin to appear in Mobile Center, so we’ll move on to the next task and add in some event tracking.
2. Immediately after the **MobileCenter** line of code added previously, add the line of code below. It’s a relatively straightforward request to track a text event.

Import Analytics class to index js file:

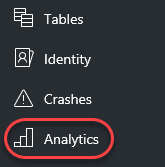
import Analytics from "mobile-center-analytics";

And call it in render function of main component:

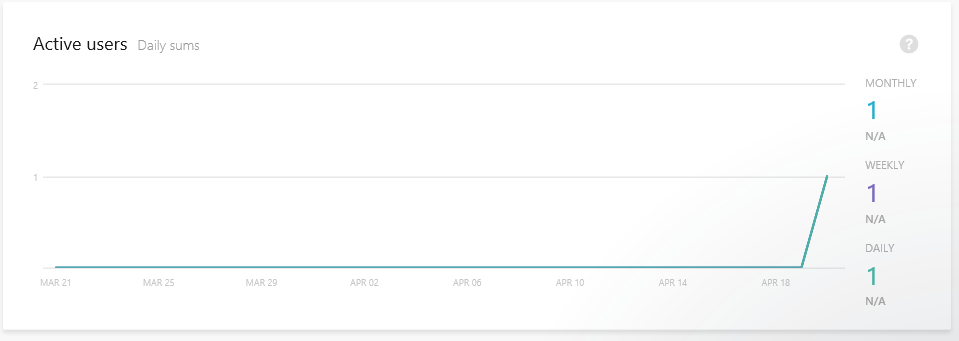
render() {

Analytics.TrackEvent("Session was started");

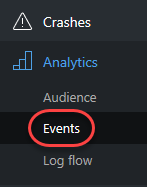
1. Go back to Simulator and press R two times to reload application. Keep in mind that while we’re using Mobile Center for an iOS/React Native application, Mobile Center supports every major platform out there. As a result, you should check out the guidance for integrating other platform combinations with the Mobile Center events guidance at <https://docs.microsoft.com/en-us/mobile-center/analytics/understand-events>.
2. Return to the Mobile Center browser window. Select the **Analytics** tab.



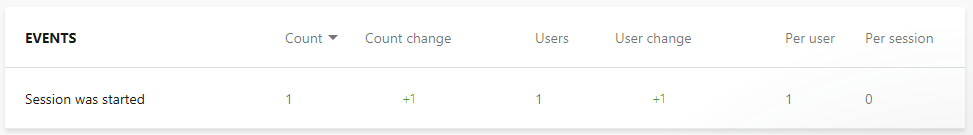
1. By now, the first user session should have been processed for display. If not, try refreshing the window every few seconds until it appears. Note that you can scroll down the page to get insight about the sessions, devices, geographies, and languages of your users, as well as reporting of the breakdown of devices per version.



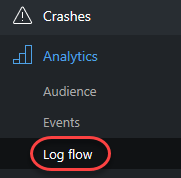
1. Select **Analytics | Events**.



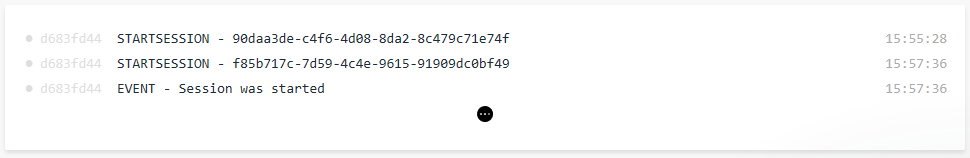
1. If the events haven’t started showing up yet, try refreshing the browser. If you run the application multiple times, then the count should increment each time you do.



1. Select **Analytics | Log flow**.



1. Log Flow will show you in real time what is being received by the backend. While the transmission may be delayed by a few seconds, this is a great place to get immediate notifications about sessions tarts, crashes, and events.



# Summary

Congratulations on completing this Quick Start Challenge! In this lab, you’ve learned how to use Mobile Center to instrument and monitor your mobile apps in development and production.

# Additional Resources

If you are interested in learning more about this topic, you can refer to the following resources:

**Documentation**: <https://docs.microsoft.com/en-us/mobile-center/>

**GitHub SDK**: <https://github.com/Microsoft/mobile-center-sdk-react-native>

**Team blog**: <https://blogs.msdn.microsoft.com/visualstudio/tag/visual-studio-mobile-center/>